第 1 章复习要点

一．计算机发展历史

1. 从第一代到第四代计算机各自的主要特点

第一代采用电子管元件，十进制。

第二代采用晶体管元件，磁芯作内存，磁鼓、磁带作外存等。

第三代采用中小规模集成电路，半导体存储器作内存，出现了微程序控制，Cache，虚拟存储器，流水线等技术。**IBM 公司提出了“兼容机”的概念，DEC 公司提出了总线结构。**

第四代采用大规模/超大规模集成电路，出现了微处理器，出现了共享存储器，分布式存储器及大规模并行处理系统等技术。

2.冯诺依曼**结构要点**；“存储程序”思想。

将事先编好的程序和原始数据送入主存中；启动执行后，在不需要操作人员干预下，自动完成逐条取出指令和执行指令的任务。

二．计算机系统的基本组成

1. 现代计算机结构模型的基本构成及其执行程序（指令序列）的步骤
2. 指令与数据
3. 软件与硬件的接口界面---ISA 指令集体系结构
4. 系统软件与应用软件的概念

三．计算机系统的层次结构

1. 现代计算机系统中从硬件、ISA，到操作系统、语言处理系统和应用程序的层次结构。
2. 最终用户、应用程序员、系统管理员、系统程序员分别工作的层面。
3. ISA 涉及的主要内容

四． 计算机系统性能评价

1. 响应时间，吞吐率，CPU 时间，MIPS，MFLOPS，基准程序（Benchmarks）

相应时间（response time）（执行时间或等待时间）：指从作业提交开始到作业完成所用的时间

吞吐量（throughput）（带宽）：指单位时间内所完成的工作量

CPU 时间：指 CPU 真正花在执行该程序的时间

1. 用户 CPU 时间：用来运行用户代码的时间
2. 系统 CPU 时间：为了执行用户程序而需要运行操作系统程序的时间

专门用来进行性能评价的一组程序

2. CPI 的计算

3.性能评价的工具——基准程序（Benchmarks）

4.阿姆达尔定律及简单应用

名词解释：

系列机（兼容机）：相同或相似的指令集，相同或相似的操作系统的计算机

中央处理器 CPU：整个计算机的核心部件，主要用于指令的执行

数据通路：由操作元件和存储元件通过总线方式或分散方式连接而成的进行**数据传送、处理和存储** 的路径。

对齐：**要求数据的地址是相应的边界地址**

第 2 章数据的机器级表示 复习要点

一． 数值数据的表示

1. 定点数的表示

（1）二进制原码、补码的表示必须搞清楚；移码表示：标准移码的偏置常数 2^{n-1}，它与补码的关系：最高位相反，其余位相同。

（2）无符号数表示和带符号数表示进行数据长度扩展时的差异。

2. 浮点数表示 IEEE754 标准

（1）单精度 32 位和**双精度 64 位的格式**；偏置常数的取值。

（2）规格化数的阶码取值范围；规格化尾数的表示（隐含位 1）；IEEE754 浮点数几个特殊数据的表示形式：0、∞、NaN(非数)、非规格化数。

（3）用 ASCII 码表示十进制数：前分隔数字串和后嵌入数字串两种格式表示正负号；用 BCD 码表示十进制数：正负数的表示方法，位数不等于 8Bit 的整数倍时需补 0。

二． **非数值数据的表示**

汉字的编码：输入码、内码、字模点阵码，它们分别有哪些常用编码。

三． 数据的宽度，存储和排列顺序

1.概念：字、字长、最低有效字节 LSB、最高有效字节 MSB

2.数据按字节存储时，多字节数据的地址涉及到数据是大端方式还是小端方式。

3. 指令存放时大端和小端只影响指令中的多字节常数，不影响其他字段的存放顺序。

4. 数据存储时存在边界对齐和不对齐问题，它们在存储空间和访问速度上存在差异

四． 数据的检错与纠错

了解数据检错与纠错的基本原理，知道常用的几种数据校验编码：奇偶校验码、海明校验码和循环冗余校验码。

第 3 章运算方法和运算部件 复习要点

一．串行进位加法器与并行进位加法器

1.并行进位加法器比串行进位加法器速度快的原因。

2.全先行进位加法器、局部先行进位加法器和多级先行进位加法器的区别。

二．ALU 的构成

1. 整数加减运算器的基本构成（关键:如何实现减法运算）；

2. ALU 如何控制实现加、减、与、或等等各种功能；

3. **ALU 的 OF、SF、CF 和 ZF 标志信息如何产生。**

4. 如何判断无符号数和带符号数加减运算时发生溢出。

三．定点数的加减乘法运算方法

1. **补码、原码、移码的加减运算方法；**

2. 标准移码与 IEEE754 移码的加减运算方法的差别；

3. 无符号数乘法的机器实现基本步骤；

4. 无符号数乘法的硬件逻辑结构；

5.原码一位乘法机器实现的基本原理。

四．浮点数运算

1. 浮点数加减运算的对阶原则和方法；

2. 如何计算移码表示的阶码的和与差（标准移码与 IEEE754 移码有什么差别）；

3. 如何计算一个移码数减 1

4. 尾数规格化中的右规和左规方法；

5. **尾数的舍入处理常用方法；**

6. 如何判断结果溢出（上溢和下溢）。

第 4 章指令系统 复习要点

一、指令系统设计

1. 指令中应该包括哪些字段？提供操作数或操作数地址有哪些方式？
2. 明白指令格式设计的几个基本原则。
3. 掌握一般计算机中 8 种基本寻址方式产生操作数的过程。
4. 掌握变长操作码编码方法的基本原理
5. **条件码的产生；条件码的应用：无符号数的大小比较方法和带符号数的比较方法。**
5. 了解堆栈型、累加器型、通用寄存器型和装入/存储型四种指令格式风格操作过程和特点。掌握 CISC 和 RISC 两类指令的特点。了解 CISC 指令系统的 2/8 规律。

最常使用的是一些简单指令，这些指令占程序的 80%，但只占指令系统的 20%。而且在微程序控制的计算机中，占指令总数 20%的复杂指令占用了控制存储器容量的 80%。

二、程序的机器级表示

1. MIPS 指令有哪些寻址方式？有哪些特有的寻址方式？掌握 MIPS 的基本汇编指令表示。
2. 掌握 MIPS 计算机的 R 型、I 型和 J 型三种指令格式。

（1）清楚 RR 型指令、移位指令、寄存器跳转指令三种类型指令格式的区别和特点；R 型指令采用了什么寻址方式？

（2）I 型指令可能会出现寄存器寻址、立即数寻址、相对寻址和偏移寻址（基址或变址）几种寻址方式，这些寻址方式在每条指令都会出现吗？**I 型指令有哪些类型的指令？（双目运算、分支、存储访问）**，掌握分支型和存储访问型指令的地址形成方法、双目运算型指令中对立即数的扩展方法。

（3）J 型指令中会出现哪些寻址方式？掌握 J 和 Jal 两条指令的功能

3. 高级语言与汇编指令之间的转换：掌握基本的高级语言中的运算表达式 ，If 语句，循环，数组访问的汇编语言实现。

第 5 章中央处理器 复习要点

数据通路:（操作元件和存储元件通过总线方式或分散方式连接而成的进行**数据存储、处理、传送**的路径。）

数据通路的功能：进行数据存储、处理、传送。

一、单周期数据通路的设计

1. 操作元件和存储元件的概念，单周期 MIPS 计算机中有哪些操作元件和存储元件？

2. 寄存器和寄存器组、理想存储器的读过程和写过程，以及它们的区别。

读是组合逻辑，写为时序逻辑

3. 熟练掌握课件中的最基本的 7 条指令执行时数据通路中信息的流动过程，以及在取指令部件中的信息处理，包括元件的连接和所需要的各种控制信号的取值等。**同时也能够在前述基本结构上扩展指定功能和格式的一些指令。**

二、单周期控制器的设计

1. 运算器的功能是如何控制的？掌握指令译码的基本原理，OP 和 func 字段如何与指令功能对应？明白每个控制信号与指令译码的对应关系。

2. 单周期 CPU 的周期长度是由什么指令、哪些因素决定的？通过分析每条指令的执行过程，明确所传输和处理的信息的流动路径，**然后根据各个部件的延时（操作元件的延迟或存储元件的三个时间 setup time、hold time 和 clock to Q time）算出指令执行所必需的最短时间。**

三、微程序控制原理（背诵）

1. 微程序控制器的基本思想

微程序控制器的基本思想：

- **仿照程序设计的方法，编制每条指令对应的微程序**
- **每个微程序由若干条微指令构成，各微指令包含若干条微命令**
- **所有指令对应的微程序放在只读存储器中，执行某条指令就是取出对应微程序中的各条微指令，对微指令译码产生对应的微命令(即控制信号)。**
- **这个只读存储器称为控制存储器（Control Storage），简称控存CS 。**

2. 比较硬连线控制器和微程序控制器的优缺点。

硬连线控制器的特点：

优点：速度快，适合于简单或规整的指令系统，例如，MIPS指令集。

缺点：它是一个多输入/多输出的巨大逻辑网络。对于复杂指令系统来说，结构庞杂，实现困难；修改、维护不易；灵活性差。甚至无法用有限状态机描述！

3. 指令、微程序、微指令、微命令、微操作它们之间的**关系**

4. 了解水平型微指令和垂直型微指令的概念

基本思想：相容微命令尽量多地安排在一条微指令中。

基本思想：一条微指令只控制一、二个微命令。

四、异常和中断处理（背诵）

1. 异常和中断（外部）的区别

内部“异常”：在 CPU 内部发生的意外事件或特殊事件

外部“中断”：在 CPU 外部发生的特殊事件，通过“中断请求”信号向 CPU 请求处理。

类别	原因	异步/同步	返回行为
中断 Interrupt	来自I/O设备、其他硬件部件	异步	总是返回到下一条指令
陷入Trap	有意识安排的	同步	返回到下一条指令
故障Fault	可恢复的错误	同步	返回到当前指令
终止Abort	不可恢复的错误	同步	不会返回

2. 掌握计算机中对异常/中断的软件识别（MIPS 计算机）和硬件识别这两种不同方式的基本过程。

有两种不同的识别方式：软件识别和硬件识别（向量中断方式）。

（1）软件识别（MIPS采用）

设置一个异常状态寄存器（MIPS中为Cause寄存器），用于记录异常原因。操作系统使用一个统一的异常处理程序，该程序按优先级顺序查询异常状态寄存器的各位，识别出异常事件。

（例如：MIPS中位于内核地址0x8000 0180处有一个专门的异常处理程序，用于检测异常的具体原因，然后转到内核中相应的异常处理程序段中进行具体的处理）

（2）硬件识别（向量中断）（80x86采用）

用专门的硬件查询电路按优先级顺序识别异常，得到“中断类型号”，根据此号，到中断向量表中读取对应的中断服务程序的入口地址。

第 6 章 指令流水线简介

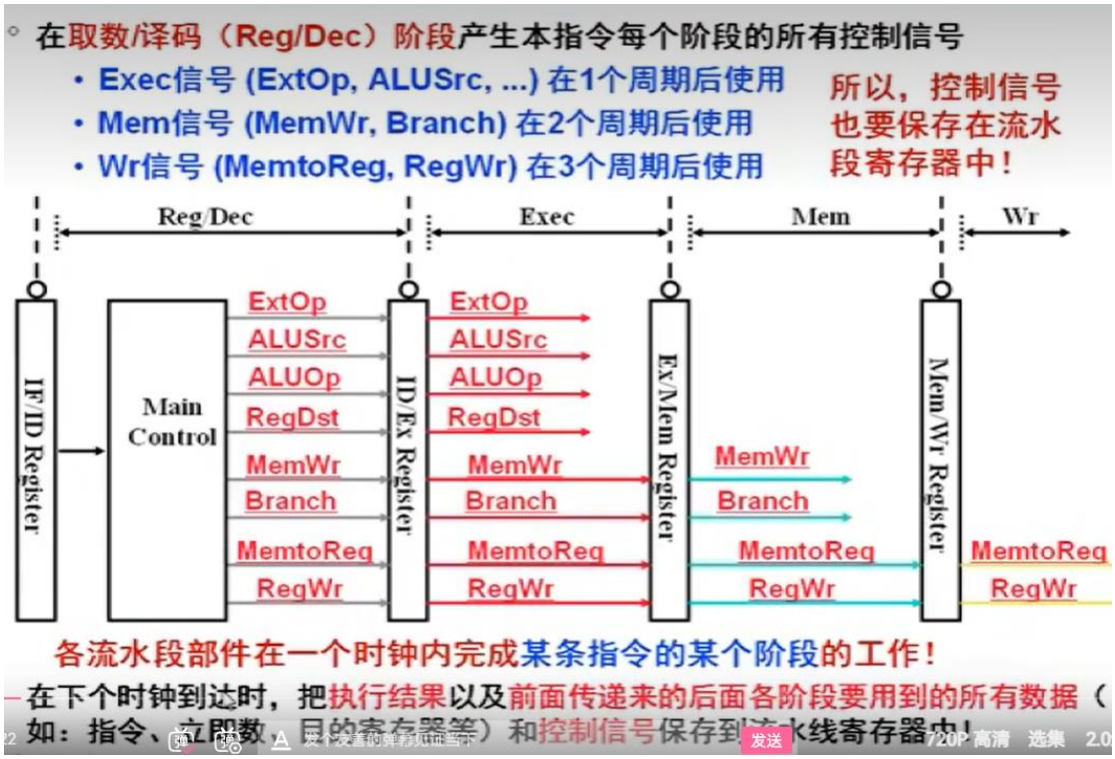
1.取指；取数/译码；计算（执行）；读存储器；写寄存器

2.控制信号

计算（执行）：Extop，ALUop, ALUsrc，RegDst

访存：MemWr，Branch

写寄存器：MemtoReg,RegWr



通过对前面流水线数据通路的分析，得知：

- PC需要写使能吗？ 每个时钟都会改变PC，故不需要!
- 流水段寄存器需要写使能吗？ 每个时钟都会改变流水段寄存器，故不需要!
- Ifecth阶段和Dec/Reg阶段都没有控制信号
- Exec阶段的控制信号有四个
  - ExtOp (扩展器操作)：1- 符号扩展；0- 零扩展
  - ALUSrc (ALU的B口来源)：1- 来源于扩展器；0- 来源于BusB
  - ALUOp (主控制器输出，用于辅助局部ALU控制逻辑来决定ALUCtrl)
  - RegDst (指定目的寄存器)：1- Rd；0- Rt
- Mem阶段的控制信号有两个
  - MemWr (DM的写信号)：Store指令时为1，其他指令为0
  - Branch (是否为分支指令)：分支指令时为1，其他指令为0
- Wr阶段的控制信号有两个
  - MemtoReg (寄存器的写入源)：1- DM输出；0- ALU输出
  - RegWr (寄存器堆写信号)：结果写寄存器的指令都为1，其他指令为0

3.流水段寄存器储存的信息

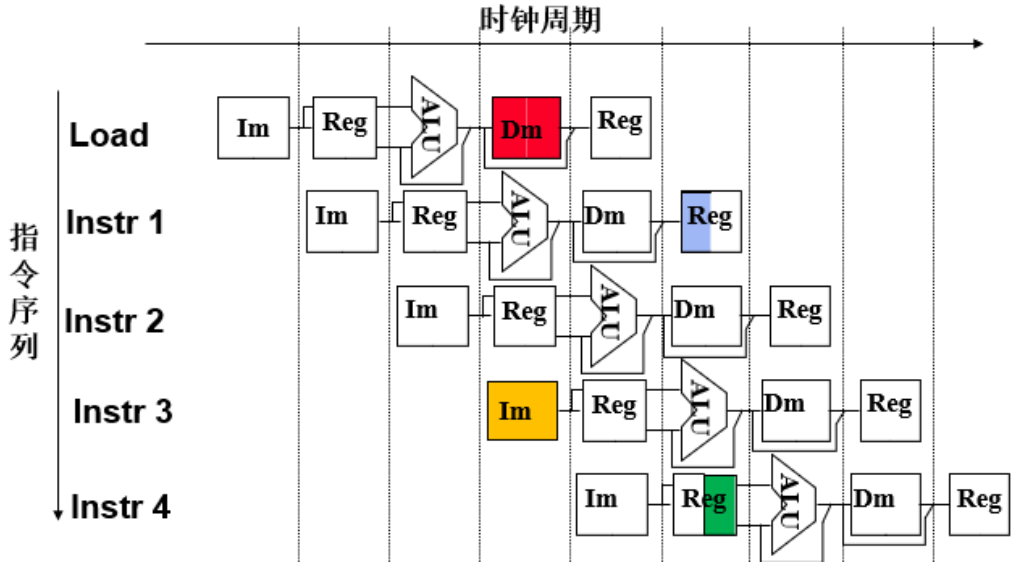
4.冒险

结构冒险：Mem，分为 IM,DM

结构冒险的解决方法

为了避免结构冒险，规定流水线数据通路中功能部件的设置原则为：

- **每个部件安排在特定的阶段使用！ 如：ALU总在指令的第三阶段使用**
- **将Instruction Memory (Im) 和 Data Memory (Dm)分开**
- **将寄存器读口和写口独立开来**



数据冒险；

方法 1：硬件阻塞（stall）

方法 2：软件插入“NOP”指令(编译时)

方法 3：合理实现寄存器组的读/写操作

方法 4：转发（Forwarding 或 Bypassing 旁路）技术（只能解决 ALU 出来的数据，MEM 不行）

方法 5：编译优化--调整指令顺序（不能解决所有数据冒险）

控制冒险；

现象：当遇到改变指令执行顺序的转移指令（调用、返回等）、异常和中断情况时，在形成转移目的地址之前，流水线中已取了后续指令并在执行，这时就需要清除流水线中的部分指令的执行。

- **控制逻辑的输出（控制信号）在ID阶段生成，并存放在ID/EX流水段寄存器中，然后每来一个时钟跟着指令传送到下一级流水段寄存器**

复习要点

一、流水线数据通路和控制

1. 单周期指令模型与流水模型的性能比较：

单周期指令模型下时钟周期如何确定？流水模型下时钟周期如果确定？

为什么流水线方式下单条指令执行时间不能缩短，但能大大提高指令吞吐率？

2. 具有哪些特征的指令集有利于流水线执行？（四个特征）

长度尽量一致，有利于简化取指令和指令译码操作

格式少，且源寄存器位置相同，有利于在指令未知时就可取操作数

load / Store 指令才能访问存储器，有利于减少操作步骤，规整流水线
内存中”对齐”存放，有利于减少访存次数和流水线的规整

3. 能对常见的 7 条指令的各个流水阶段的划分及其所使用的功能部件情况进行分析。

4. 对于五阶段流水线数据通路，能分析 7 条指令执行时的各流水段寄存器存储了哪些信息。

二、流水线冒险处理（背诵）

1. 什么是流水线冒险，它分为哪些类型？

在指令流水线中，当遇到某些情况使得流水线无法正确执行后续指令，而引起流水线阻塞或停顿，这个现象称为流水线冒险

2. 结构冒险的现象是什么？如何处理结构冒险？

现象：同一个部件同时被不同指令所用，即使用硬件资源时发生了冲突。

3. 数据冒险的现象是什么？对给出一个指令序列，能分析会在哪里产生了冒险现象。

现象：后面指令用到前面指令结果时，前面指令的结果还没产生。

4. 掌握常见的硬件阻塞（stall）、软件插入“NOP”指令、合理实现寄存器组的读/写操作、转发（Forwarding 或 Bypassing 旁路）技术、编译优化--调整指令顺序这些处理方法的基本原理。

5. 控制冒险的现象是什么？了解常见的四种处理方法。

现象：当遇到改变指令执行顺序的转移指令（调用、返回等）、异常和中断情况时，在形成转移目的地址之前，流水线中已取了后续指令并在执行，这时就需要清除流水线中的部分指令的执行。

第 7 章存储器分层体系结构 复习要点

基本术语：

记忆单元 （存储基元 / 存储元 / 位元）（Cell）具有两种稳态的能够表示二进制数码 0 和 1 的物理器件。

存储单元 / 编址单位（Addressing Unit）具有相同地址的位构成一个存储单元，也称为一 个编址单位。

存储体/ 存储矩阵 / 存储阵列（Bank）所有存储单元构成一个存储阵列。 编址方式（Addressing Mode）字节编址、按字编址。

存储器地址寄存器（Memory Address Register - MAR）用于存放主存单元地址的寄存器。

存储器数据寄存器（ Memory Data Register-MDR (或 MBR) ）用于存放主存单元中的数据的寄存器。

一、存储器概述和存储器芯片

1. 熟悉随机存取存储器、顺序存取存储器、直接存取存储器、相联存储器、

只读存储器、读写存储器、非易失（不挥发）性存储器、易失（挥发）性存储器、静态存储器、动态存储器这些名称的含义。

2. 层次结构存储系统中的寄存器、高速缓存、内存(主存)、外存它们所在的位置、工作速度、存储容量、成本等的相对大小和大致的数量级。这些存储器和前述各类存储器之间的对应关系。

3. 静态存储器和动态存储器的基本工作机制；动态存储器刷新的概念，按行刷新的含义。最大刷新周期的确定的依据是什么。

DRAM 的集中刷新、分散刷新和异步刷新的刷新操作与正常访存分别是如何安排的？

4. 了解 SDRAM 芯片中的突发传输方式

二、存储器容量的扩展及其与 CPU 的连接

位扩展、字扩展、字位扩展方式，系统存储容量的计算，芯片数的计算；这几种扩展方式下的芯片(组)与片选信号的地址线分配；各芯片(组)的地址范围的计算、划分。片选信号用地址信号表示的逻辑表达式。

多模块存储器：连续编址，交叉编址

三、高速缓冲存储器(cache)

1. 直接映射、全相联映射、组相联映射三种方式的映射关系；三种方式下的主存地址与 cache 的行、内容之间的对应关系；cache 容量的计算方法，注意区分数据区、标记、有效位。

2. CPU 对 cache 的访问时，直接映射采用的是按地址进行查找的方法，而全相联映射采用的是用多个比较器进行同时比对查找到 cache 的行；组相联映射则结合了上述两种方法，即由地址查找到组，再对组内的各行“标记”用多个比较器进行同时比对。和相联存储器的概念有什么关系？

3. 三种映射方式中哪些需要替换算法？了解“先进先出 FIFO”和“最近最少用 LRU”替换算法。了解写策略中的命中和未命中的处理方式。

四、虚拟存储器

1. 虚拟存储器的基本思想---分页的基本思想

（1）页（虚页、逻辑页）与页框（实页、物理页）的概念

（2）逻辑地址（虚拟地址）与物理地址（实地址、主存地址）的概念
（3）虚拟存储器机制由硬件与操作系统共同协作实现。

- 逻辑地址转换为物理地址是由硬件(CPU 中的存储管理部件)完成，其中虚页号到页框号的转换是通过查表（页表或快表）实现。
- 当发生程序或数据访问缺页时，由操作系统在主存和磁盘之间进行信息交换。为什么不直接用硬件来处理？
- 页表由操作系统管理维护

（4）为什么页的大小比 cache 映射中的主存块要大得多？
（5）页框与虚拟页之间采用全相联映射，为什么不采用其他映射方式？
（6）在处理页框与虚拟页的一致性问题时采用回写（write back）方式，为什么不采用全写方式？
2. 页表的基本结构
（1）页表项中的装入位、修改位、存放位置的含义。为什么页表项中没有虚页号？
（2）如何区分未分配页、已分配的缓存页和已分配的未缓存页。
3. 快表—TLB
（1）快表存储在什么地方？采用快表的目的是什么？
（2）**快表与页表之间一般采用组相联或全相联映射，**
（3）快表的表项由页表项内容+标记(tag)，在全相联和组相联映射方式下标记字段分别是什么内容？
4. 理解根据虚拟地址，通过 TLB、页表、cache，最终转换为实地址的查找转换过程。

第 8 章互连及输入输出组织

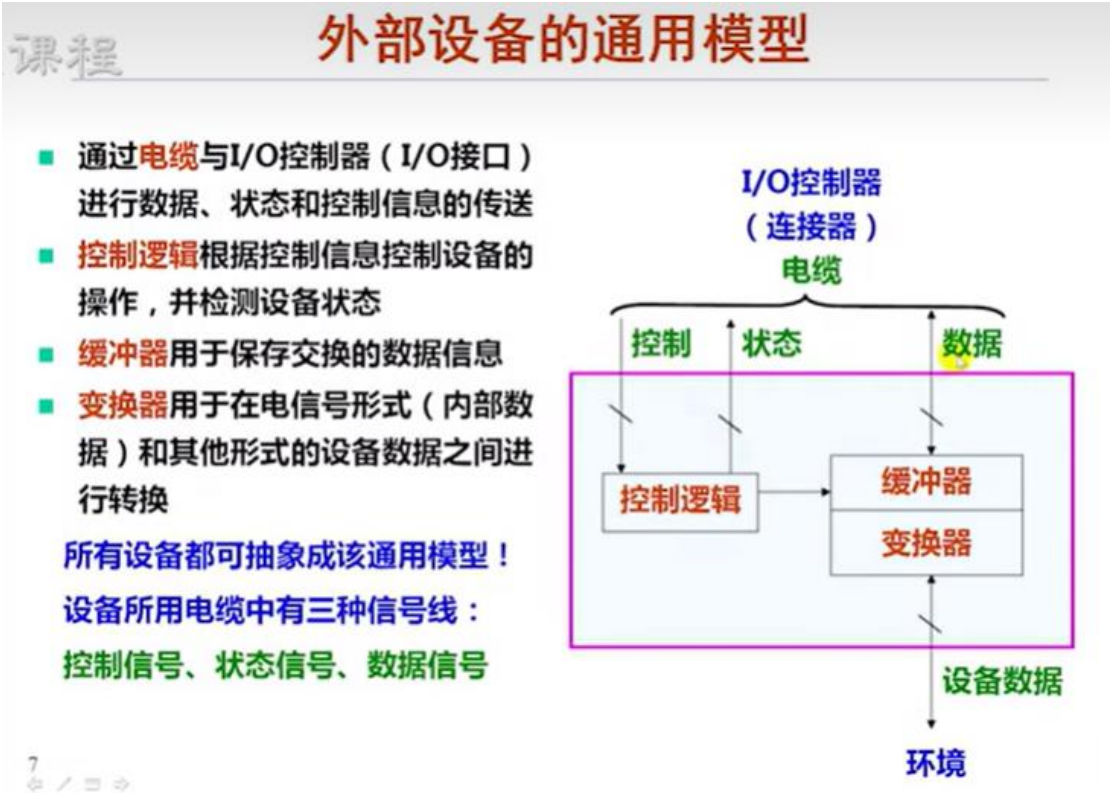
复习要点

全考默写加理解，无计算

一、I/O 系统及 I/O 设备
1. I/O 系统性能指标：吞吐率（I/O 带宽）和响应时间的定义

2. 外设的通用模型（抽象模型）以及各部分的作用

- **当外设准备好数据后，向 DMA 控制器发 DMA 请求信号，DMA 控制器再向 CPU 发总线请求，CPU 让出总线后，由 DMA 控制器控制总线进行传输，无需 CPU 干涉**



3. 磁盘上数据定位（地址）：磁道号、磁头号、扇区号；磁盘数据的存取以块（扇区）为单位。磁盘操作包括：寻道操作、旋转等待操作和读写操作。

4. 低密度磁盘：各个磁道上的扇区数相同，每个磁道存储的数据量相同，内磁道的位密度比外磁道高；高密度磁盘：各个磁道上的位密度相同，各磁道存储的数据量不同，外磁道上的扇区数比内磁道多。高密度磁盘比低密度磁盘容量高很多。

5. 磁盘容量：未格式化容量和格式化容量，它们的计算方法。

6. 硬盘的主要技术指标：平均存取时间 T 及其计算

T = 平均寻道时间 + 平均旋转等待时间 + 数据传输时间

7. 冗余磁盘阵列 RAID 的基本思想；RAID 的 3 个特性；RAID 不同级别的含义：表示具有上述 3 个特性的不同设计结构。

8. 固态硬盘 SSD 采用 NAND 闪存芯片代替磁盘作为存储介质；具有比磁盘速度快很多；类似磁盘以区块写入和抹除的方式进行数据的读取和写入；其写操作比读操作慢很多；早期的 SSD 与主机的接口与磁盘兼容，现在的 SSD 采用了更快的 PCI-E 或 M.2 接口。

二、总线及系统互连、I/O 接口

1. 总线的定义；

- **总线是指在各种层次上提供部件之间的连接和交换信息的通路**

芯片内总线，系统总线，通信总线

理解系统总线、通信总线、处理器总线、存储总线、I/O 总线、

总线宽度、总线带宽、总线工作频率等概念。

现在总线设计的趋势是：**点对点、异步、串行**

FSB、QPI 和 PCI-Express x n 总线的特点，它们的带宽计算方法。

（1）CPU 总线

前端总线（Front Side Bus，FSB）

并行传输、同步定时方式

QPI 的一个时钟周期传两次

（2）IO 总线

第一代：ISA/EISA 总线、VESA 总线，早被淘汰

第二代：PCI、AGP、PCI-X，被逐渐淘汰

第三代：PCI-Express

PCI-Express 总线（串行总线，主流总线）

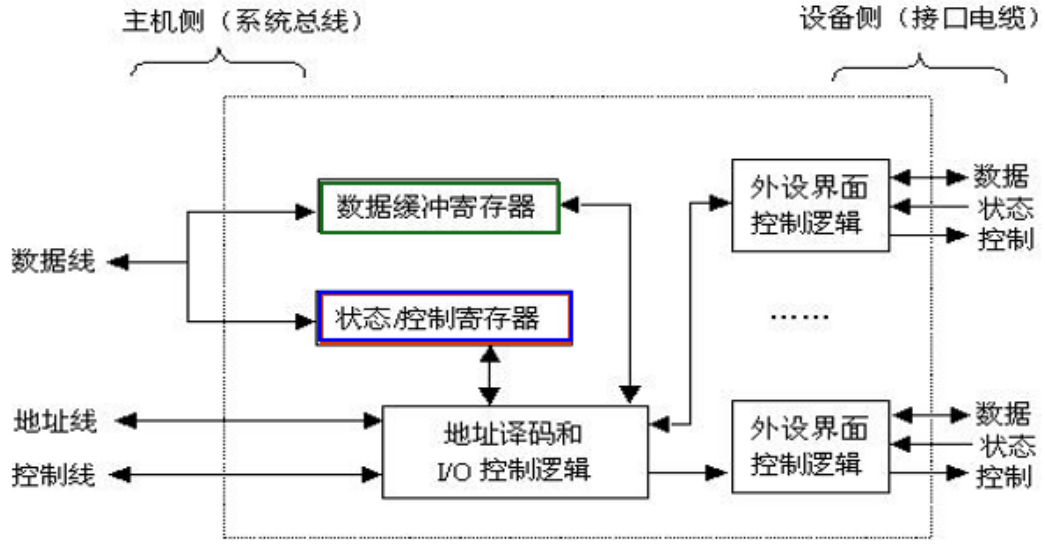
2. I/O 接口的 **5 大功能**；

I/O 接口的通用结构中包含的几个寄存器及其作用。

什么是 I/O 端口？**将 I/O 控制器中 CPU 能够访问的各类寄存器称为 I/O 端口**

I/O接口（I/O控制器）的结构

- **不同I/O模块在复杂性和控制外设的数量上相差很大**
- **I/O控制器的基本结构：**



通过发送命令字到I/O控制寄存器来向设备发送命令

通过从状态寄存器读取状态字来获取外设或I/O控制器的状态信息

通过向I/O控制器发送或读取数据来和外设进行数据交换

将I/O控制器中CPU能够访问的各类寄存器称为I/O端口

对外设的访问通过向I/O端口发命令、读状态、读/写数据来进行

3. I/O 端口的两种编址方式：独立编址方式和统一编址方式的含义和特点。

独立编址需要专门的 IO 指令

三、I/O 数据传送控制方式

1. **程序直接控制方式、程序中断方式和 DMA 方式**的各自工作过程、特点，以及性能比较分析。

2. 中断的概念；

中断与异常的区别；

问题：中断响应的时点与异常处理的时点是否相同？为什么？

通常在一条指令执行结束后开始查询有无中断请求，有的话立即响应，所以，一般在指令执行完时响应中断；而“异常”发生在指令执行过程中，所以，不能等到指令执行完才进行异常处理。

中断过程：**中断响应和中断处理；中断响应的过程；中断处理的过程；**

中断响应是指主机发现外部中断请求，中止现行程序的执行，到调出中断服务程序这一过程。

(1) 中断响应的条件

① CPU处于开中断状态

② 在一条指令执行完

③ 至少要有一个未被屏蔽的中断请求

① 关中断

0 => 中断允许触发器 C_{IEN}

② 保护断点和程序状态

PC => 堆栈（或特殊寄存器 EPC）

PSW => 堆栈

③ 识别中断源

中断处理是指执行相应中断服务程序的过程。

单重中断和多重中断的差别及如何实现。

单重中断:不允许在中断处理时被新的中断打断，因而直到中断返回前才会开中断。单重中断系统无需设置中断屏蔽字。

多重中断的概念：在一个中断处理（执行中断服务程序）过程中，若有新的中断请求发生， 且新中断优先级高于正在执行的中断，则中止正在执行的中断服务程序，转去处理新的中断。这种情况为多重中断，也称中断嵌套。

差别：一个允许被打断、一个不允许被打断；一个无屏蔽字，一个有屏蔽字

3. 中断响应优先级和中断处理优先级的概念；如何通过中断屏蔽字来调整中断处理优先级。

中断响应优先级----由查询程序或硬联排队线路决定的优先权，反映多个中断同时请求时选择哪个响应。

中断处理优先级----由各自的中断屏蔽字来动态设定，反映本中断与其它中断间的关系。

4. DMA 方式的基本要点；

- 与中断控制方式结合使用
  - DMA 传送前，“寻道”“旋转”等操作结束时，通过“中断”告知 CPU。
  - 在 DMA 控制器控制总线进行数据传送时，CPU 执行其他程序。
  - DMA 传送结束时，要通过“DMA 结

束中断”告知 CPU。

DMA 常用的三种方式：**CPU 停止法、周期挪用法、交替分时访问法**及其优缺点。

5. DMA 控制器的功能；DMA 操作的三个步骤；

①. DMA 控制器的预置（初始化）----软件实现

②. DMA 数据传送------硬件实现

③. DMA 结束处理------软件实现

DMA I/O 传送方式与中断 I/O 传输方式的差异比较。